Python Developer

# 

[**Introduction**](#_1fob9te) **1**

[**Tasks**](#_3znysh7) **1**

[FizzBuzz](#_2et92p0) 1

[Requirements](#_tyjcwt) 1

[Input](#_3dy6vkm) 1

[Output](#_1t3h5sf) 2

[Example:](#_4d34og8) 2

[Valuation service](#_2s8eyo1) 2

# Introduction

Here are few programming tasks to check your programing skills. Please store your code in some private repo (for example BitBucket)

# Tasks

## FizzBuzz

Write a program that prints the integers from n to m (inclusive).

### Requirements

* for multiples of three, print Fizz (instead of the number)
* for multiples of five, print Buzz (instead of the number)
* for multiples of both three and five, print FizzBuzz (instead of the number)

1 <= n < m <= 10000

### Input

Two numbers in two lines (n, m)

### Output

One result per line including requirments

### Example:

Sample input:

3

16

Sample output:

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

16

## Valuation service

You are building a valuation service.

On the input you've got 3 files containing:

\* data.csv - product representation with price,currency,quantity,matching\_id

\* currencies.csv - currency code and ratio to PLN, ie. GBP,2.4 can be converted to PLN with procedure 1 PLN \* 2.4

\* matchings.csv - matching data matching\_id,top\_priced\_count

Now, read all the data. From products with particular matching\_id take those with the highest total price (price \* quantity), limit data set by top\_priced\_count and aggregate prices.

Result save to top\_products.csv with four columns: matching\_id,total\_price,avg\_price,currency, ignored\_products\_count.

Unit tests are necessary.

-----------------------------------------------------------

currencies.csv

currency,ratio

GBP,2.4

EU,2.1

PLN,1

data.csv

id,price,currency,quantity,matching\_id

1,1000,GBP,2,3

2,1050,EU,1,1

3,2000,PLN,1,1

4,1750,EU,2,2

5,1400,EU,4,3

6,7000,PLN,3,2

7,630,GBP,5,3

8,4000,EU,1,3

9,1400,GBP,3,1

matchings.csv

matching\_id,top\_priced\_count

1,2

2,2

3,3